METHOD OF BUS PRIORITY ARBITRATION

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the priority benefit of U.S.A provisional application serial no. 60/225,018, filed August 11, 2000, and Taiwan application serial no. 90110586, filed May 3, 2001.

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The invention relates to a peripheral component interconnect (PCI) bus. More particularly, the present invention relates to the operation of a PCI bus priority arbitration.

Description of the Related Art

[0002] Referring to FIG. 1, a structure diagram schematically illustrates a computer architecture that uses a conventional Periphery Component Interconnect (PCI) bus. A central processing unit 10 and a memory 11 are coupled with a PCI bus 14 via a host bridge 12. The PCI bus 14 in turn is coupled with a plurality of PCI-compatible devices such as a graphic adapter 16a, an expansion bus bridge 16b, a network interface 16c, and a SCSI interface 16d.

[0003] Each PCI-compatible devices has a master. Before using the PCI bus 14, each of the masters outputs a request signal to request the use of the PCI bus 14. The bus arbiter embedded within the host bridge 12 outputs a grant signal to the master as a response to the request of the master.

[0004] The data transaction between the different PCI-compatible devices (or between host bridge and PCI-compatible master) can be performed via interface control signals as described hereafter. The initiator outputs a cycle frame signal (FRAME#) to initiate a data transaction. The cycle frame signal (FRAME#) implies that data transaction through the PCI bus has started. When the cycle frame signal (FRAME#) remains at a low level, data transaction is still under way. At the same time, a valid address is put on the address/data bus lines (AD) during a so-called "address phase". Meanwhile, valid bus commands (satisfying PCI specifications) are outputted from a plurality of command/byte enable (CBE [3:0]) lines to inform the target about the type of data transaction requested by the initiator. After the output of valid address, the requested data is put on the address/data bus line (AD) during a so-called "data phase". In the meantime, byte enable signals corresponding to the encoded valid bus command are delivered through the CBE lines for data transaction.

[0005] When the initiator terminates the cycle frame signal (FRAME#), either the last set of data is outputted, or the data transaction is completed. During the data transaction, an initiator ready signal (IRDY#) and a target ready signal (TRDY#) work with each other to indicate that the initiator and the target are ready to process the data transaction. During a data read operation, putting up an initiator ready signal (IRDY#) implies that the initiator is ready to receive data. During a data write operation, the activation of a target ready signal (TRDY#) means that the target device is ready to receive data. In addition, there is a stop signal (STOP#) that can be set up by the target to inform the initiator to terminate the current data transaction.

[0006] Referring to FIG. 2, a block diagram schematically illustrates a general PCI architecture comprising a plurality of PCI-compatible devices. These PCI-

compatible devices (38, 40, 42, 44) are coupled to a PCI bus 36. An bus arbiter 22 embedded in a host bridge 20 is coupled with the PCI bus 36 to give the PCI compatible devices authorization to use the PCI bus 36.

[0007] Each of PCI-compatible devices has a master, which can assert a request signal (REQ) to the bus arbiter 22 for use of the PCI bus 36 at any time. The bus arbiter 22 then can assert a grant signal (GNT) to one of the PCI-compatible devices to grant the use of the PCI bus 36.

[0008] However, if the data requested by a request master is not ready, the bus arbiter 22 outputs a response signal (STOP#) to the PCI-compatible device to signal that the grant for using the PCI bus is refused. The master of the PCI-compatible device then deasserts its request signal (REQ#). At the next transaction cycle, the master of the PCI-compatible device outputs a request signal (REQ#) to the bus arbiter 22 again. The above sequence of transactions is repeated until the data is ready. Then, when the bus arbiter 22 authorizes the PCI-compatible device to use the PCI bus 36, the data requested by the PCI-compatible device can be effectively transferred.

[0009] Referring to FIG. 3, a diagram schematically illustrates the state of response in the conventional operation of a PCI system. GNT0(50), GNT1(52), GNT2(54), and GNT3(56) respectively represent the different states of response of the bus arbiter with respect to each of the masters. At one particular state of response, the bus arbiter only responds to the request of one master. At a starting stage, the bus arbiter is positioned at the initial state of response 50 (GNT0). The state of response 52 (GNT1) consequently has the highest priority when the bus arbiter will shift to the next state of response. That means that no matter from which master the following request signal is delivered, the bus arbiter will identically follow the path g01 to first

shift to the state of response 52 (GNT1). Similarly, when the bus arbiter is at the state of response 52 (GNT1), the state of response 54 (GNT2) has the highest priority. When the bus arbiter is at the state of response 54 (GNT2), the state of response 56 (GNT3) has the highest priority, etc.

[0010] By arranging the states of response within a closed circular pathway as in the above-described, the request from each of the masters will be cyclically processed by the bus arbiter cyclically passing from one state of response to another. However, because the bus arbiter identically runs along the same direction, a delay of the response might occur as described hereafter. For example, let's assume that the bus arbiter is at the state of response 52 (GNT1) addressed to one particular master and that the master of the state of response 54 (GNT2) outputs a request signal. In accordance with the priority rule, the bus arbiter shifts to the state of response 54 (GNT2) via the pathway g12. However, the data requested by the master of the state of response 52 (GNT1) has not been completed yet. The master of the state of response 52 (GNT1) thus must wait for the bus arbiter to perform an entire cycle via the successive states of response 54 (GNT2), 56 (GNT3), and 50 (GNT0) to the state of response 52 (GNT1) before obtaining the right to use the PCI bus again.

[0011] Referring to FIG. 4, a time/sequence diagram schematically illustrates the cyclical response sequence of a conventional PCI structure. For the sake of simplification, FIG. 4 illustrates the example of only two requests cyclically outputted by a first and second masters. At the transaction time cycle 60, both the first master and the second master request (REQ1# and REQ2# staying in low level) for the grant to use the PCI bus. In accordance with the priority rule, the bus arbiter first responds to the request from the first master (GNT1#). The first master outputs a cycle frame

signal (FRAME#) and an initiator ready signal (IRDY#) meaning that the first master is in a state of waiting for the data transaction. The bus arbiter then outputs a device select signal (DEVSEL#) and a stop signal (STOP#), wherein the stop signal (STOP#) implies that the data is not ready, and then the request signal (REQ1) and the grant signal (GNT1#) change to high level. Because the first master does not receive any data, the first master then asserts a request signal (REQ1#) meaning that the first master is still requesting the use of the PCI bus. It should be noted that the PMADS signal represents the address into which the host bridge must transfer the data, while the PMRDY signal indicates whether the data is ready to be transferred.

[0012] At the following transaction time cycle 62, the bus arbiter of the host bridge shifts to the request of the second master(REQ2#) in accordance with the priority rule. The operation sequences that take place are similar to those of the first master, and the data for the second master also is not ready. In accordance with the priority rule, the bus arbiter then shifts to the request of the first master(REQ1#) at the following transaction time cycle 64. At the transaction time cycle 64, the bus arbiter refuses to grant the use of the PCI bus to the first master by outputting the stop signal (STOP#) because the data is not ready. However, soon after the request signal (REQ1#) is changed to high level subsequent to the stop signal (STOP#), the PMRDY signal is outputted, announcing that the data is ready. However, the bus arbiter still shifts to the request of the second master(REQ2) at the following time cycle 66 in accordance with the priority rule. The data transaction for the PCI-compatible device of the first master thus is processed only at the transaction time cycle 68, once the bus arbiter shifts again to the state of response addressed to the first master.

[0013] The above description reveals that the conventional response operation performed by the bus arbiter disadvantageously generates a delay in the data transfer to the PCI-compatible devices.

SUMMARY OF THE INVENTION

[0014] A major aspect of the present invention is thus to provide a method of bus priority arbitration driven by data. It means that the bus arbiter does not follow the conventional priority rule. The bus arbiter gives a highest priority to a master to use the PCI bus if this master's data is ready. Utilizing the method, the conventional delays of data transfer on the PCI bus can be eliminated.

[0015] To accomplish at least the foregoing objectives, the method of the PCI bus arbitration comprises the following steps. First, the requests of each masters is evaluated according to a predefined orderly rotation. The evaluation of the requests of the masters according to the predefined orderly rotation is stopped when a data requested by one of the master is ready. Then, attribute the highest priority to the master which the data is ready for the grant to use the bus. The data then is effectively transferred via the bus. Finally, once the data transfer is completed, the evaluation of the requests of the master according to the predefined orderly rotation resumes. **[0016]** To accomplish at least the foregoing objectives, the present invention further provides a peripheral device interconnect structure that comprises a bus, a host bridge connected to the bus, and a plurality of peripheral devices connected to the bus, each of the peripheral devices embedding a master. The host bridge is arranged such that it can:

[0017] evaluate a plurality of request information from each of the master according to a predefined orderly rotation;

[0018] receive information indicating that a data transfer for one of the master controllers is ready; and

[0019] stop evaluating the request information from each of the masters according to the predefined orderly rotation when receiving information indicating that a data transfer is ready to attribute the highest priority to one of the peripheral devices for granting use to the bus.

[0020] It is to be understood that both the foregoing general description and the following detailed description are exemplary, and are intended to provide further explanation of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] The accompanying drawings are included to provide a further understanding of the invention, and are incorporated in and constitute a part of this specification. The drawings illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention. In the drawings,

[0022] FIG. 1 is a block diagram schematically illustrating an example of a PCI structure in a conventional computer;

[0023] FIG. 2 is a block diagram schematically illustrating a general PCI structure;

[0024] FIG. 3 is a state of response diagram schematically illustrating the conventional operation of a PCI structure;

[0025] FIG. 4 is a time/sequence diagram schematically illustrating an example of the conventional operation of a PCI structure;

[0026] FIG. 5 is a state of response diagram schematically illustrating the operation of a PCI structure according to a preferred embodiment of the present invention; and

[0027] FIG. 6 is a time/sequence diagram schematically illustrating the operation of a PCI structure according to a preferred embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0028] The following detailed description of the embodiments and examples of the present invention with reference to the accompanying drawings is only illustrative and not limiting.

[0029] Referring to FIG. 5, a state of response diagram schematically illustrates the operation of a PCI structure based on the data according to a preferred embodiment of the present invention. In comparison with the operation shown in FIG. 3, a major feature of the present invention is that the grant signals from the bus arbiter is not cyclically processed. Instead, the bus arbiter can shift from one state of response to any of the other states of response according to a priority evaluated from the requested data.

[0030] Referring to FIG. 5 and FIG. 1, the operation of the present invention now is described as applied to the hardware structure shown in FIG. 1. The state of response 70 (GNT0) indicates a response from the bus arbiter of the host bridge 12 to the request of the master of the graphic adapter 16a. The state of response 72 (GNT1) indicates a response to the expansion bus bridge 16b, and the state of response 74

(GNT2) indicates a response to the network interface 16c. The state of response 76 (GNT3) indicates a response to the SCSI interface 16d.

[0031] Let's assume that the bus arbiter of the host bridge 12 is positioned at the state of response 70 (GNT0) at a starting stage. The master of the PCI-compatible device 16b (expansion bus bridge) outputs a request signal. The bus arbiter then shifts to the state of response 72 (GNT1) via the pathway 0A according to, for example, a predefined anti-clockwise rotation. Meanwhile, the host bridge 12 accesses the data stored in the memory 11 requested by the PCI-compatible device 16b. When the master of the PCI-compatible device 16c (network interface) outputs a request signal, the bus arbiter shifts to the state of response 74 (GNT2) via the pathway 1A according to the predefined anti-clockwise rotation. While shifting to the state of response 74 (GNT2), if the data requested by the PCI-compatible device 16b is ready, the host bridge 12 attributes the highest priority to thePCI-compatible device 16b. The bus arbiter consequently shifts back to the response state 72 (GNT1) via the pathway 2C. In the present invention, the data thus drives the priority selection for granting use of the PCI bus. The PCI-compatible device 16b then can receive the data by using the PCI bus 14. After the data transaction is completed, the bus arbiter can either return to the state of response from which it shifted when the access priority was changed, or shift to the next state of response according to the predetermined anti-clockwise rotation.

[0032] Now let's assume that in the above example the PCI-compatible device 16d (SCSI interface) also outputs a request signal when the bus arbiter has shifted to the response state 74 (GNT2). The bus arbiter then shifts to the state of response 76 (GNT3) via the pathway 2A. The data for the PCI-compatible device 16b is ready only after the bus arbiter is at the state of response 76 (GNT3). The host bridge 12

then attributes the highest priority of use the PCI bus to the PCI-compatible device 16b. Thus, the bus arbiter directly shifts to the state of response 72 (GNT1) via the pathway 3B to grant the use of the PCI bus 14 to the PCI-compatible device 16b. After the data transfer to the PCI-compatible device 16b via the PCI bus has been completed, the bus arbiter then shifts to either the state of response 76 (GNT3) via the pathway 1B or the state of response 74 (GNT2) via the pathway 1A.

[0033] Referring to FIG. 6, a time/sequence diagram schematically illustrates a bus priority arbitration based on data according to a preferred embodiment of the present invention. For the sake of simplification, the time/sequence diagram illustrates the operation of the present invention only for two requests from two masters. However, it will be apparent from the description hereafter that the present invention can be effectively implemented for PCI bus that comprises more than two PCI-compatible devices.

[0034] At the transaction time cycle 80, a first master and a second master output request signals (REQ1# and REQ2#). The bus arbiter of the host bridge (not shown), for example, first responds to the request signal (REQ1#) of the first master (GNT1#). At this time, the first master outputs a cycle frame signal (FRAME#) and an initiator ready signal (IRDY#) meaning that it is in a state of waiting for the data transaction. The bus arbiter of the host bridge then outputs a device select signal (DEVSEL#) and a stop signal (STOP#), wherein the stop signal (STOP#) implies that the data is not ready, and then the request signal (REQ1) and the grant signal (GNT1#) change to high level. Because the first master does not receive any data, the first master then asserts a request signal (REQ1#) that activates the state of response GNT1#, meaning that the request from the first master has not been satisfied yet.

[0035] At the following transaction time cycle 82, the bus arbiter responds to the request signal (REQ#) from the second master in accordance with the priority rule. Similar to the above, the data requested by the second master is not ready and the second master can not use the PCI bus because of the stop signal (STOP#).

[0036] At the following transaction time cycle 84, the bus arbiter responds to the request signal (REQ1#) from the first master in accordance with the priority rule. The data still is not ready and the bus arbiter outputs a stop signal (STOP#) to change the request signal (REQ1#) to high level. However, while the request signal (REQ1#) is outputted subsequent to the stop signal (STOP#), the PMRDY signal is activated, implying that the data is ready. Unlike the conventional operation of the structure described in FIG. 4, the data is taken into account by the host bridge that consequently attributes the highest priority to the first master. At the next transaction time cycle 86, the bus arbiter thus grants use of the PCI bus to the first master to effectively transfer the data via the PCI bus.

[0037] In conclusion, in the method of the present invention, when a master outputs a request signal for using the PCI bus, the host bridge informs the corresponding data storage device to prepare the requested data. Once the data is ready, the host bridge consequently attributes the highest priority to the master regardless of which state of response the bus arbiter of the host bridge is in. The data thus drives the priority arbitration in the present invention, which advantageously eliminates the conventional delay of data transfer.

[0038] It will be apparent to those skilled in the art that various modifications and variations can be made to the structure of the present invention without departing from the scope or spirit of the invention.